Docket No.: MWS-077RCE

Application No.: 09/911,819

AMENDMENTS TO THE CLAIMS

What is claimed is:

1. (Currently Amended) In a computing device, a method comprising:
providing a definition of a function associated with a first language; ~~and~~
creating description information about the function from the definition of a function
associated with a first language; and ~~, wherein the description information enables translation of
a call to the function in the first language into a call to a corresponding function in a second
language without requiring processing of the definition of the function.~~
translating a call to the function in the first language into a call to a corresponding
function in a second language using the description information, without processing the
definition of the function.

2. (Original) The method of claim 1, further comprising: storing the description information in a
file of description items.

3. (Previously Presented) The method of claim 1, wherein creating description information about
the function comprises: examining the definition of the function associated with the first
language; and deriving information about the function.

4. (Cancelled)

5. (Currently Amended) The method of claim 4 1, further comprising: storing a translated
function in the second language in a library of entries.

6. (Previously Presented) The method of claim 1, in which creating description information
about the function comprises: deriving a number of declared formal inputs to the function.

7. (Previously Presented) The method of claim 1, in which creating description information
about the function comprises: deriving a number of declared formal outputs to the function.

BEST AVAILABLE COPY

Docket No.: MWS-077RCE

Application No.: 09/911,819

8. (Previously Presented) The method of claim 1, in which creating description information about the function comprises: deriving a scope of the function.

9. (Previously Presented) The method of claim 1, in which creating description information about the function comprises: determining whether the function accepts a variable number of arguments.

10. (Previously Presented) The method of claim 1, in creating description information about the function comprises: determining whether the function returns a variable number of results.

11. (Previously Presented) In a computing device, a method comprising: providing a file of description items, each item including description information about a function associated with a first language, wherein the description information enables translation of a call to the function in a first language into a call to a corresponding function in a second language without requiring processing of the definition of the function; and using the file of description items to translate a first program file into a second program file.

12. (Original) The method of claim 11, wherein the description information about the function comprises: a descriptor identifying a declared number of formal inputs to the function.

13. (Original) The method of claim 11, wherein the description information about the function comprises: a descriptor identifying a declared number of formal outputs to the function.

14. (Original) The method of claim 11, wherein the description information about the function comprises: a descriptor identifying a scope of the function.

15. (Original) The method of claim 11, wherein the description information about the function comprises: a descriptor identifying an acceptance of a variable input argument list into the function.

16. (Original) The method of claim 11, wherein the description information about the function comprises: a descriptor identifying a return of a variable output argument list from the function.

17. (Previously Presented) The method of claim 11, wherein using the file of description items comprises: for each call to a function in the first program file, retrieving an item from the file of description items; using the description information in the item to translate the call to the function in the first language into a call to a corresponding function in the second language; and storing the translated call in the second program file.

18. (Original) The method of claim 11, wherein using the file of description items comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying an acceptance of a variable input argument list into the function.

19. (Original) The method of claim 11, wherein using the file of description items comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying a return of a variable output argument list from the function.

20. (Original) The method of claim 11, wherein using the file of description items comprises: generating a call through a normal interface for the function if the description information includes a descriptor identifying a known number of input and output arguments to the function.

21. (Previously Presented) In a computing device, a method comprising: providing a library file including functions defined by a first language; creating a function library and a description file from the library file, the function library including one or more functions defined by a second language, each function in the function library being a translated version of a function in the library file, and the description file including description information about each function in the library file, wherein the description information enables translation of a call to the function in the first language into a call to a corresponding function in the second language without requiring processing of the definition of the function; and using the description file to translate a program file from the first language into the second language, wherein each call in the program file to a function in the library file is translated into a call to a corresponding function in the second language.

A

22. (Previously Presented) The method of claim 21, wherein creating a function library comprises: translating the call to each function in the library file into a call to a corresponding function in the second language.

23. (Previously Presented) The method of claim 21, wherein creating a creating description file comprises: examining the definition of each function in the library file; and deriving information about each function.

24. (Previously Presented) The method of claim 23, further comprising: using the derived information about each function to create description information; and creating a description file including description information about each function in the library file.

25. (Original) The method of claim 21, wherein using the description file comprises: for each call in the program file to a function in the library file, retrieving the description information about the function from the description file; and using the description information to translate the call to the function in the first language into a call to a corresponding function in the second language.

26. (Original) The method of claim 21, wherein using the description file comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying an acceptance of a variable input argument list into the function.

27. (Original) The method of claim 21, wherein using the description file comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying a return of a variable output argument list from the function.

28. (Original) The method of claim 21, wherein using the description file comprises: generating a call through a normal interface for the function if the description information includes a descriptor identifying a known number of input and output arguments to the function.

5

Application No.: 09/911,819

29. (Currently Amended) A computer program product, tangibly stored on a computer-readable medium, for creating a data file, the product comprising instructions operable to cause a programmable processor to: obtain a definition of a function associated with a first language; and create description information about the function from the definition of the function associated with a first language; wherein the description information enables translation of a call to the function in the first language into a call to a corresponding function in a second language without requiring processing of the definition of the function and translate a call to the function in the first language into a call to a corresponding function in a second language using the description information, without processing the definition of the function.

30. (Original) The product of claim 29, further comprising instructions operable to cause a programmable processor to: store the description information in a file of description items.

31. (Previously Presented) The product of claim 29, wherein creating description information comprises: examining the definition of the function associated with the first language; and deriving information about the function.

32. (Original) The product of claim 31, further comprising instructions operable to cause a programmable processor to: use the derived information to create the description information.

33. (Cancelled)

34. (Previously Presented) The product of claim 29, in which creating description information comprises: deriving a number of declared formal inputs to the function.

35. (Previously Presented) The product of claim 29, in which creating description information comprises: deriving a number of declared formal outputs to the function.

36. (Previously Presented) The product of claim 29, in which creating description information comprises: deriving a scope of the function.

37. (Previously Presented) The product of claim 29, in which creating description information

6

comprises: determining whether the function accepts a variable number of arguments.

38. (Previously Presented) The product of claim 29, in which creating description information comprises: determining whether the function returns a variable number of results.

39. (Previously Presented) A product, stored on a machine-readable medium, for translating a program file, the product comprising instructions operable to cause a processor to: provide a file of description items, each item including description information about a function associated with a first language, the description information enabling translation of a call to the function into a call to a corresponding function in a second language without requiring processing of the definition of the function; and use the file of description items to translate a first program file into a second program file.

40. (Original) The product of claim 39, wherein the description information about the function comprises: a descriptor identifying a declared number of formal inputs to the function.

41. (Original) The product of claim 39, wherein the description information about the function comprises: a descriptor identifying a declared number of formal outputs to the function.

42. (Original) The product of claim 39, wherein the description information about the function comprises: a descriptor identifying a scope of the function.

43. (Original) The product of claim 39, wherein the description information about the function comprises: a descriptor identifying an acceptance of a variable input argument list into the function.

44. (Original) The product of claim 39, wherein the description information about the function comprises: a descriptor identifying a return of a variable output argument list from the function.

45. (Previously Presented) The product of claim 39, wherein using the file of description items comprises: for each call to a function in the first program file, retrieving an item from the file of description items; using the description information in the item to translate the call to the

7

function in the first language into a call to a corresponding function in the second language; and storing the translated call in the second program file.

46. (Original) The product of claim 39, wherein using the file of description items comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying an acceptance of a variable input argument list into the function.

47. (Original) The product of claim 39, wherein using the file of description items comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying a return of a variable output argument list from the function.

48. (Original) The product of claim 39, wherein using the file of description items comprises: generating a call through a normal interface for the function if the description information includes a descriptor identifying a known number of input and output arguments to the function.

49. (Previously Presented) A computer program product, tangibly stored on a computer-readable medium, for translating function calls, the product comprising instructions operable to cause a programmable processor to: provide a library file including functions defined by a first language; create a function library and a description file from the library file, the function library including one or more functions defined by a second language, each function in the function library being a translated version of a function in the library file, and the description file including description information about each function in the library file, wherein the description information enables translation of a call to the function in the first language into a call to a corresponding function in the second language without requiring processing of the definition of the function; and use the description file to translate a program file from the first language into the second language, wherein each call in the program file to a function in the library file is translated into a call to a corresponding function in the second language.

50. (Previously Presented) The product of claim 49, wherein creating a function library comprises: translating the call to each function in the library file into a call to a corresponding

function in the second language.

51. (Original) The product of claim 49, wherein creating a description file comprises: examining the definition of each function in the library file; and deriving information about each function.

52. (Original) The product of claim 51, further comprising: using the derived information about each function to create the description information; and creating a description file including description information about each function in the library file.

53. (Original) The product of claim 49, wherein using the description file comprises: for each call in the program file to a function in the library file, retrieving the description information about the function from the description file; and using the description information to translate the call to the function in the first language into a call to a corresponding function in the second language.

54. (Original) The product of claim 49, wherein using the description file comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying an acceptance of a variable input argument list into the function.

55. (Original) The product of claim 49, wherein using the description file comprises: generating a call through a function evaluation interface for the function if the description information includes a descriptor identifying a return of a variable output argument list from the function.

56. (Original) The product of claim 49, wherein using the description file comprises: generating a call through a normal interface for the function if the description information includes a descriptor identifying a known number of input and output arguments to the function.